

Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada *Mobile E-Parking* Berbasis Sistem Informasi Geografis

Cahyo Prianto^{1*}, Muh. Kusnadi²

^{1,2} Program Studi DIV Teknik Informatika, Politeknik Pos Indonesia, Bandung, Jawa Barat

^{1,2}Jln. Sariosih No.54 Sarijadi, Kota Bandung, 40151, Jawa Barat, Indonesia

email: ¹cahyoprianto@poltekpos.ac.id, ²muh.kusnadi99@gmail.com

Received: 11 Juni 2018; Revised: 1 Oktober 2018; Accepted: 20 Oktober 2018

Copyright ©2018, Politeknik Harapan Bersama, Tegal

Abstract – Lack of information about the position of a place and find the best route to get to a destination is complicated, especially looking for a route in finding a parking space. Unavailability of applications that provide information and navigate parking users to find the best route in realtime is a problem that must be answered. Dijkstra's algorithm is one method of solving problems in finding the best route. The Dijkstra algorithm has compared the possible routes to be traversed and calculated every possible distance. The route with the shortest distance would be the best choice This application was built using an Android-based Geographic Information System that utilizes the Google Maps feature as a parking user interface service. This study uses the Java programming language with Android Studio as an IDE (Integrated Development Environment) and the prototyping model is used in developing problem solving and using UML (Unified Modeling Language) as a system design tool. The results that have been achieved from this research is that Dijkstra's algorithm can answer the problem faced by providing the best route navigation from the user's location to the parking lot at the destination

Abstrak – Kurangnya informasi mengenai posisi suatu tempat dan menemukan rute terbaik untuk menuju sebuah tempat tujuan termasuk hal yang cukup rumit terutama mencari rute dalam menemukan tempat parkir. Tidak tersedianya aplikasi yang menyediakan informasi serta menavigasi pengguna parkir untuk menemukan rute terbaik secara realtime menjadi suatu masalah yang harus dicari jawabannya. Algoritma *Dijkstra* merupakan salah satu metode penyelesaian masalah dalam pencarian rute terbaik. Algoritma *Dijkstra* membandingkan kemungkinan rute yang akan dilalui dan dihitung setiap kemungkinan jarak. Rute dengan jarak terpendek akan menjadi pilihan terbaik. Aplikasi ini dibangun dengan menggunakan sistem informasi geografis berbasis android yang memanfaatkan fitur *Google Maps* sebagai layanan antar muka pengguna parkir. Penelitian ini menggunakan bahasa pemrograman Java dengan Android Studio sebagai IDE (*Integrated Development Environment*) dan model *prototyping* digunakan dalam pengembangan penyelesaian masalah serta menggunakan UML (*Unified Modelling Language*) sebagai *tools* perancangan system.

***) Corresponding author:** (Cahyo Prianto)

Email: cahyoprianto@poltekpos.ac.id

Hasil yang telah dicapai dari penelitian ini adalah algoritma *Dijkstra* dapat menjawab masalah yang dihadapi yakni memberikan navigasi rute terbaik dari lokasi pengguna menuju tempat parkir di lokasi tujuan.

Kata Kunci – Rute Terbaik, Algoritma *Dijkstra*, Sistem Informasi Geografis, *Android*, *Java*.

I. PENDAHULUAN

Pencarian rute terpendek ternyata menjadi salah satu perhatian dunia. Penerapannya baik pada perencanaan pembangunan, perencanaan produksi, rute tujuan pada jalan raya, rute pesawat terbang dan sebagainya [1]. Menemukan rute terpendek ternyata tidak semudah yang dibayangkan, ada beberapa hal yang perlu di perhatikan diantaranya: Jarak, waktu ataupun harga yang biasa disebut dengan biaya pengeluaran [2]. Sistem Informasi Geografis merepresentasikan data lokasi geografis berbasiskan peta [3]. Diperlukan kordinat *latitude* dan *longitude* untuk menentukan sebuah lokasi [4]. Untuk mencapai sebuah lokasi tujuan menggunakan sistem informasi geografis, nyatanya diperlukan algoritma untuk mengetahui rute terpendek antara tempat awal dan tempat tujuan [5]. Mengakses informasi yang cepat salah satunya adalah dengan menggunakan sistem informasi geografis berbasis *mobile* [6]. Dengan memanfaatkan fitur GPS untuk mendeteksi keberadaan pengguna [7], serta penempatan lokasi tempat parkir pada sistem [8]. Dengan kata lain, tersedianya informasi pada peta berupa rute ke tempat tujuan memungkinkan pengguna mendapat navigasi ke tempat tujuan [9].

Titik terlemah dalam mendapatkan informasi adalah tidak tersedianya sarana informasi yang dapat diakses dengan cepat dan mudah [10]. Dibutuhkan kualitas pelayanan yang memadai sehingga membuat pengguna lebih mudah mengakses informasi [11]. Salah satu jenis sistem informasi yang sering dipakai adalah sistem informasi geografis [12]. Informasi yang terdapat pada sistem informasi geografis berupa deskripsi, gambar dan lokasi [13] khususnya untuk pencarian tempat parkir [14]. Pengguna terkadang sulit menemukan lokasi parkir serta ketersediaan tempat parkir yang ingin dituju [15]. Poin penting yang sangat bermanfaat

bagi pengguna adalah menemukan rute terpendek untuk sampai ke lokasi [16]. Menemukan rute terdekat merupakan masalah yang fundamental [17]. Menghitung rute terpendek dibutuhkan setiap simpul dari titik awal sampai titik akhir yang terhubung kemudian dibandingkan dengan simpul lainnya [18].

Peneliti mengusulkan sebuah algoritma yang dapat menyelesaikan masalah untuk menghitung rute terpendek dari tempat pengguna ke tempat tujuan. Algoritma *dijkstra* menyediakan performa yang efisien dengan menyelesaikan perencanaan penentuan rute terdekat [19]. Terlebih dahulu menentukan kemungkinan simpul yang akan dilalui, setiap simpul memiliki titik-titik persinggahan yang memiliki nilai yang menjadi parameter perhitungan. Pemanfaatan algoritma *dijkstra* dimana menghitung setiap kemungkinan titik yang akan dilalui yang memiliki nilai yang paling kecil memiliki kemungkinan besar memiliki rute yang lebih dekat dengan yang lain [20].

II. PENELITIAN YANG TERKAIT

Dalam sebuah penelitian yang dilakukan oleh Y. Zhang, dkk yang berjudul “*When dijkstra meets vanishing point: a stereo vision approach for road detection*” [18], membahas tentang pendeteksi jalan dengan menggunakan algoritma *Dijkstra* untuk mengenali jalan yang dapat dilalui. Dengan menampilkan peta gambar jalan pada setiap jalan memiliki warna yang berbeda yakni: warna hijau menunjukkan hasil yang positif dengan kata lain jalan tersebut dapat dilalui, warna merah menunjukkan bagian jalan tersebut sudah ada objek yang menempati seperti kendaraan ataupun sejenisnya, dan warna biru merupakan bagian pinggir jalan. Tujuan dari penelitian ini berfokus pada menemukan lokasi jalan yang hilang dari berbagai peta gambar yang di ambil, peran Algoritma *Dijkstra* yakni meminimalkan biaya pada penggunaan peta gambar dimana gradien gambar abu-abu digunakan untuk menghitung biaya antara titik piksel pengambilan gambar yang memiliki kemungkinan jalan yang hilang kemudian menemukan rute terpendek pada setiap titik piksel yang terdapat pada peta gambar jalan.

Penelitian berikutnya diusulkan oleh O.V. Gnana Swathika dan S. Hemamalini yang berjudul “*Prims Aided Dijkstra Algorithm for Adaptive Protection in Microgrids*” [21], membahas tentang rekonfigurasi mikrogrid pada topologi jaringan yang bervariasi. Algoritma *Dijkstra* bertanggung jawab untuk mengidentifikasi topologi jaringan saat ini dan membantu mengidentifikasi rute terpendek dari titik yang ditentukan ke sumber operasi terdekat mengurangi beban untuk mencapai sumber operasi yang memiliki jarak yang lebih jauh pada topologi jaringan tersebut.

Penelitian berikutnya diusulkan oleh Jin-dong Zhang, dkk yang berjudul “*Vehicle routing in urban areas based on the Oil Consumption Wight-Dijkstra algorithm*” [19], membahas tentang perencanaan rute berkendara pada area perkotaan untuk meminimalkan beberapa parameter mencakup jarak, kecepatan, waktu mengemudi, arus perjalanan, serta konsumsi oli. Melalui penentuan titik awal dan titik akhir sebagai destinasi tujuan, algoritma *Dijkstra* membantu untuk mencari rute optimal yang layak dilalui pada peta secara otomatis.

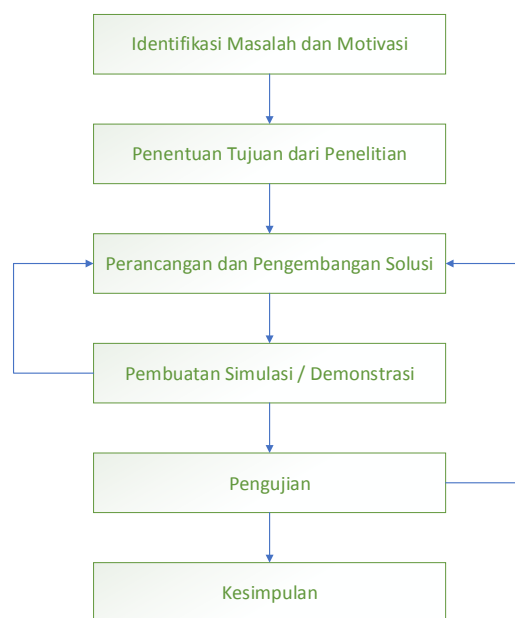
Pada penelitian sebelumnya, penggunaan algoritma *Dijkstra* umumnya banyak digunakan untuk menemukan rute

ataupun jalur optimal pada setiap objek. Perbedaan penelitian ini dengan penelitian sebelumnya yakni objek yang diteliti serta teknologi yang digunakan. Untuk mengakses informasi yang lebih cepat dan *realtime*, peneliti menampilkan informasi yang dapat diakses dengan mudah yakni dengan teknologi *mobile* berbasis android dimana dapat mempermudah pengguna baik mengakses informasi lokasi parkir dan menemukan rute terbaik untuk mencapai tujuan lokasi parkir dari titik pengguna *mobile* berada.

III. METODE PENELITIAN

A. Diagram Alur Metodologi Penelitian

Pada penelitian ini, penulis akan membuat aplikasi Sistem Informasi Geografis dan Penentuan Rute Terbaik berbasis *Android*. Untuk menyelesaikan penerapan tersebut dibutuhkan suatu metodologi penelitian yang meliputi enam tahapan yaitu identifikasi masalah dan motivasi, penentuan tujuan dari penelitian, perancangan dan pengembangan solusi, pembuatan simulasi/demonstrasi, pengujian, dan kesimpulan. Dalam pengembangan sistem ini penulis menggunakan model *prototyping*, tahapannya seperti pada Gbr.1.



Gbr.1 Diagram Alur Metodologi Penelitian

B. Algoritma Dijkstra

Algoritma yang paling banyak digunakan untuk menentukan masalah pemilihan rute, baik penentuan rute terpendek maupun rute yang paling optimal adalah algoritma *Dijkstra*. Algoritma *Dijkstra* digunakan untuk mengidentifikasi rute terpendek. Algoritma *Dijkstra* memungkinkan menemukan rute terpendek antara dua titik. Pada pemodelan kali ini, algoritma *Dijkstra* digunakan untuk membangun sistem yang mengarahkan pengguna untuk sampai ke titik tujuan. Untuk menentukan rute terpendek terlebih dahulu diidentifikasi 2 titik yang menggambarkan antara titik awal dan titik tujuan serta membuat grafik yang menggambarkan kemungkinan setiap rute yang akan dilalui, seperti yang terlihat dalam Gbr.2.

C. Simulasi Perhitungan

Tahap awal yang dilakukan yaitu membandingkan setiap kemungkinan simpul yang dilalui, setiap titik mempunyai bobot yang berbeda. Titik yang dilalui pada rute yang satu tentu saja berbeda dengan titik yang dilalui pada rute yang lainnya [6], maka dapat digunakan rumus berikut:

$$\text{ToNodeDestination} = \text{Node1} \Rightarrow \text{Node2} \Rightarrow \text{Node3} \Rightarrow \dots \Rightarrow \text{Node}_n \quad (1)$$

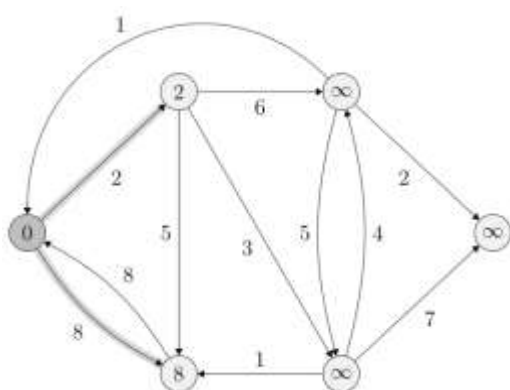
Setelah menentukan titik awal sampai ketitik tujuan, kita akan menjumlahkan setiap titik yang telah dilalui.

$$\text{getDistance} = \text{Node1} + \text{Node2} + \text{Node3} + \dots + \text{Node}_n \quad (2)$$

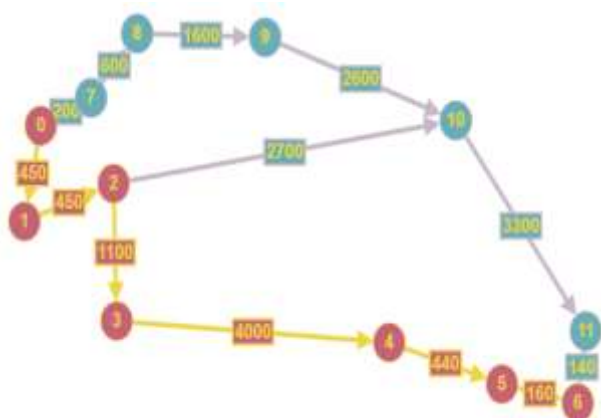
Data yang dibutuhkan merupakan data lokasi yang melakukan kerjasama dengan aplikasi sistem *e-parkir*. Untuk pengelolaan algoritma *Dijkstra* dibutuhkan data *latitude* dan *longitude* yang merepresentasikan sebuah lokasi pada fitur peta.

Pada kasus ini, peneliti melakukan simulasi pada suatu lokasi tempat parkir yakni titik awal merupakan lokasi *user* berada yaitu Jalan Sarijadi Bandung Blok 7 No.57 dan titik akhir yaitu BEC (Bandung Electronic Center) yang berada di Kota Bandung.

Tahap pertama yaitu menggambarkan grafik yang merepresentasikan titik awal ke titik tujuan. Seperti yang terlihat pada Gbr.3



Gbr. 2 Pola Kemungkinan Rute Yang Dilalui.



Gbr. 3 Simulasi Algoritma Dijkstra

Setiap perpindahan titik memiliki jarak, hal pertama yang dilakukan adalah menghitung setiap simpul jarak pada setiap kemungkinan rute yang ada. Pada Kasus ini terdapat 3 kemungkinan rute yang dapat dilalui kendaraan.

Terlihat pada Gbr. 3 tentang menentukan titik setiap rute:

Route 1 = Titik 0 => Titik 1 => Titik 2 => Titik 3 => Titik 4 => Titik 5 => Titik 6.

Route 2 = Titik 0 => Titik 7 => Titik 10 => Titik 11 => Titik 6.

Route 3 = Titik 0 => Titik 7 => Titik 8 => Titik 9 => Titik 10 => Titik 11 => Titik 6.

Menghitung jarak keseluruhan jarak masing-masing rute dalam satuan **meter**.

Route 1 = 450 + 450 + 1100 + 4000 + 440 + 160 = 6560 **meter** (6.560 km).

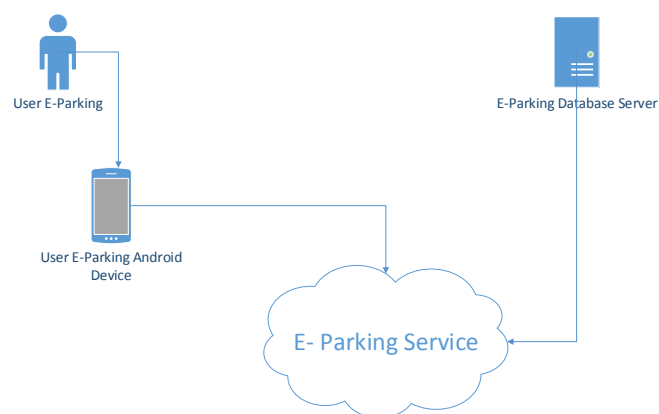
Route 2 = 450 + 450 + 2700 + 3300 + 140 = 6740 **meter** (6.740 km).

Route 3 = 200 + 600 + 1600 + 2600 + 3300 + 140 = 8440 **meter** (8.440 km).

Pada simulasi ini kita dapat mengambil kesimpulan bahwa rute terpendek jatuh pada rute pertama, karena memiliki jarak yang paling dekat untuk sampai ke titik tujuan.

D. Arsitektur Perancangan

1) Arsitektur Komputer

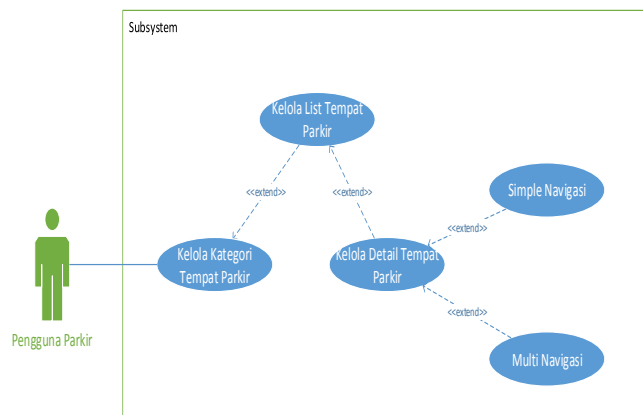


Gbr. 4 Arsitektur Komputer Perancangan E-Parking

Dalam pembangunan aplikasi *E-Parking* yang dapat diakses publik, *database* perlu di-*hosting* sehingga dapat mengakses tempat-tempat parkir. Setelah di-*hosting* maka aplikasi *E-Parking* akan mengambil API dari *database E-Parking* yang disebut juga dengan *E-Parking Service*. *E-Parking service* sebagai API selanjutnya akan diakses khusus untuk aplikasi *E-Parking* berbasis android sehingga informasi-informasi mengenai parkir dapat diakses sesuai dengan kebutuhan.

2) Use Case Diagram

Untuk merekam persyaratan fungsional sebuah system maka digunakan *usecase diagram*. *Usecase diagram* mendeskripsikan interaksi antara para pengguna sistem dengan sistem itu sendiri, dengan memberikan sebuah narasi tentang bagaimana sistem tersebut digunakan. Pada Gbr.5 penjelasan yang terkait dengan *use case* tersaji dalam Tabel 1.



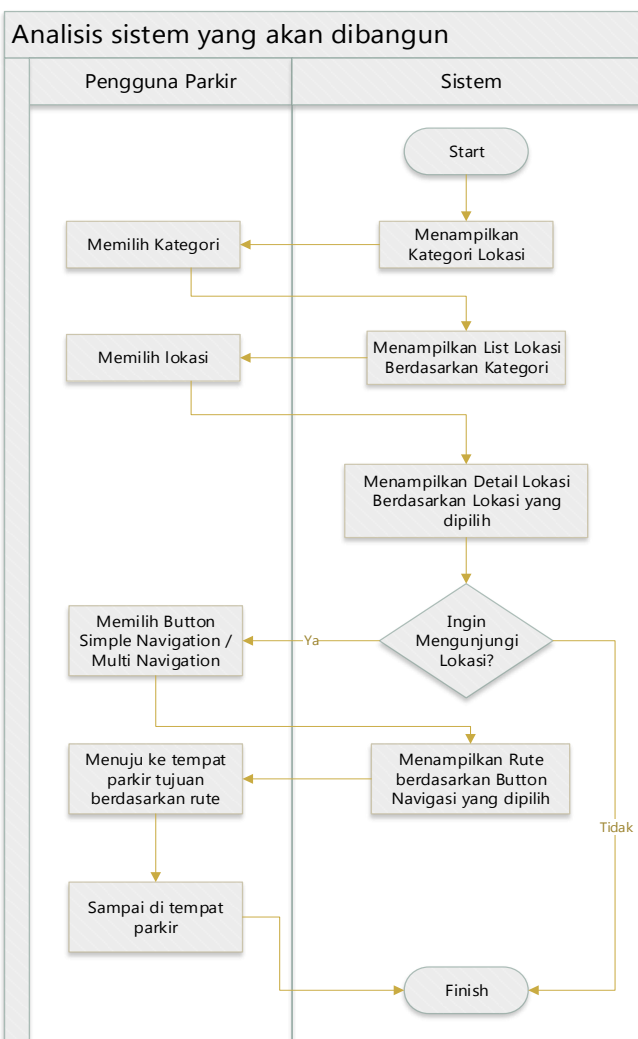
Gbr. 5 Usecase Diagram

TABEL I
DEFINISI USECASE

No	Usecase	Keterangan
1	Kelola Kategori Tempat Parkir	Melakukan pengelolaan Kategori parkir, yakni memilih salah satu kategori parkir yang tersedia
2	Kelola List Tempat Parkir	Melakukan pengelolaan List tempat parkir, yakni memilih salah satu list tempat parkir yang tersedia
3	Kelola Detail Tempat Parkir	Melihat detail tempat parkir
4	Simple Navigasi	Memilih opsi simple navigate untuk menampilkan rute terbaik
5	Multi Navigasi	Memilih opsi multi navigasi untuk menampilkan kemungkinan rute lain yang dapat dilalui

3) Flowmap

Terdapat 1 aktor dalam penerapan sistem yang akan dibangun, yaitu pengguna Parkir, sebagai *user* yang mencari tempat parker, dilihatkan pada Gbr.6.



Gbr. 6 Flowmap Sistem yang akan Dibangun

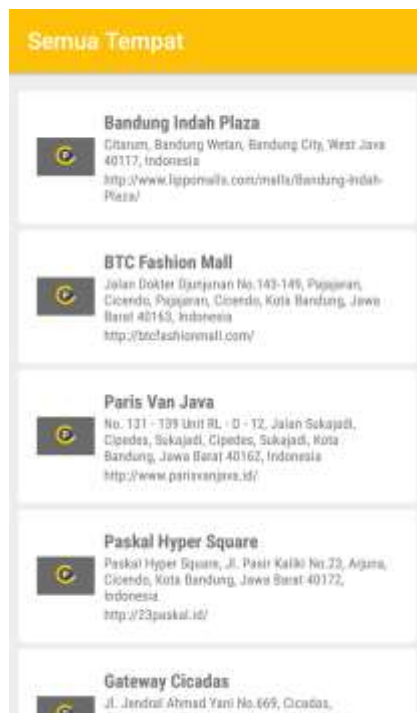
Sesuai dengan Gbr. 6, adapun *flowmap* sistem yang akan dibangun, yaitu diantaranya:

- Pengguna parkir membuka aplikasi parkir.
- Aplikasi akan menampilkan kategori-kategori tempat parkir yang tersedia. Kategori tempat parkir adalah tempat-tempat umum yang biasa dikunjungi masyarakat seperti tempat perbelanjaan (mall), apartement, sekolah dan lain sebagainya, seperti yang terlihat dalam Gbr.7.
- Pengguna memilih tempat kategori.
- Aplikasi akan menampilkan list lokasi berdasarkan kategori, terlihat pada Gbr. 8.
- Pengguna memilih lokasi yang tersedia.
- Aplikasi akan menampilkan detail lokasi yang dipilih oleh *user*. Sebagai contoh lokasi yang dipilih adalah lokasi perbelanjaan Bandung Indah Plaza seperti yang terlihat dalam Gbr. 9.
- Pengguna akan memiliki opsi apakah pengguna hanya melihat detail parkir atau memilih untuk pergi ke tempat parkir. Terdapat 2 opsi yang tersedia untuk menavigasi pengguna yaitu *Simple Navigate*: yang memperlihatkan rute terbaik yang bisa dilalui untuk menuju ke tempat parkir, dan *Multi Navigate*: yang memperlihatkan opsi rute yang lain.

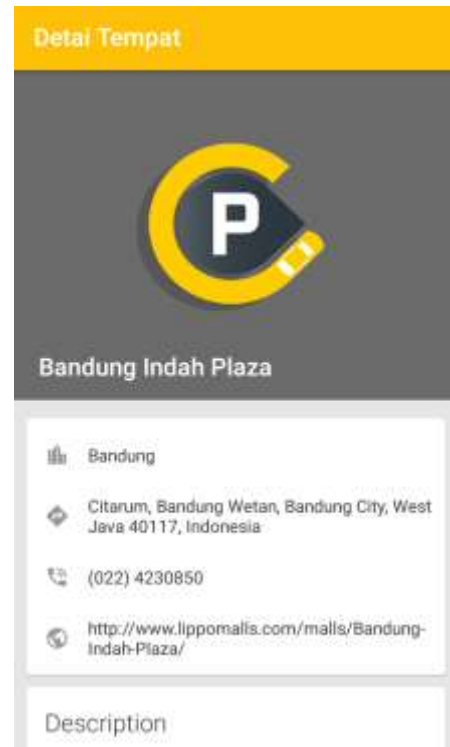
- Ketika pengguna telah memilih rute yang mana yang akan dilalui, pengguna akan menuju ke lokasi tujuan. Seperti yang disimulasikan terlihat dalam Gbr. 11 dan Gbr. 12.



Gbr. 7 Kategori Tempat Parkir



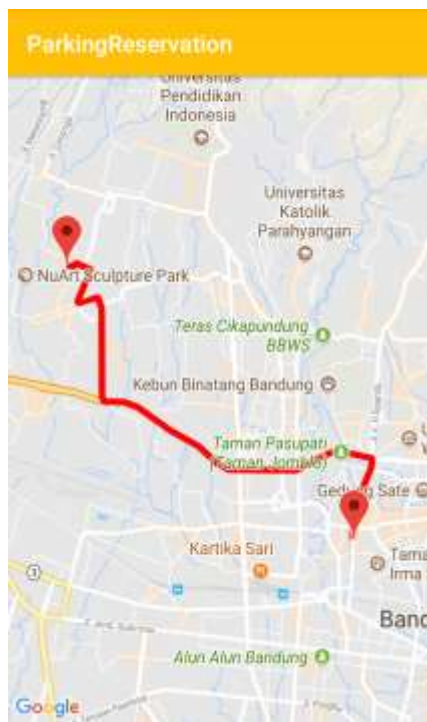
Gbr. 8 Daftar Tempat Parkir Berdasarkan Kategori yang Dipilih



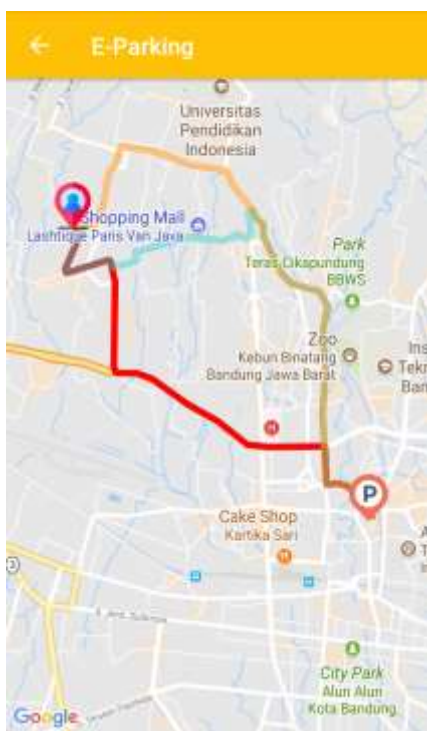
Gbr. 9 Detail Tempat Parkir Bagian 1



Gbr. 10 Detail Tempat Parkir Bagian 2



Gbr. 11 Penerapan Simple Navigasi untuk Pencarian Satu Rute Mobile E-Parking

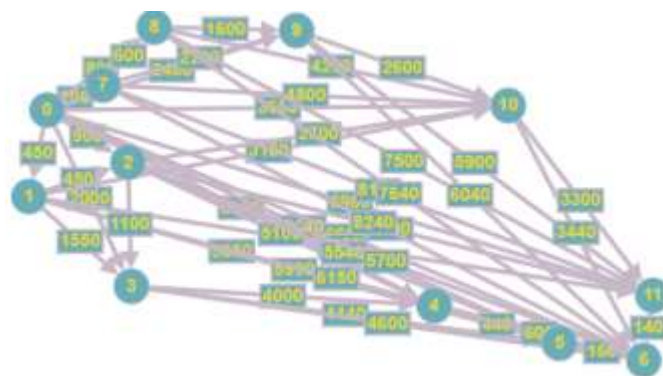


Gbr. 12 Penerapan Multi Navigasi untuk Pencarian Rute Terbaik Mobile E-Parking

IV. HASIL DAN PEMBAHASAN

Untuk menguji efektifitas algoritma Dijkstra, maka peneliti melakukan perbandingan dengan melakukan uji coba efektifitas pengaksesan rute yang lebih simpel. Peneliti

membandingkan algoritma Dijkstra yang terlihat pada Gbr.3 dengan algoritma sejenis bernama algoritma Floyd-warshall yang disajikan pada Gbr.13.



Gbr. 13 Simulasi Algoritma Floyd-warshall

Dengan membandingkan gambar tersebut, dapat terlihat bagaimana algoritma Floyd-warshall lebih mengambil setiap kemungkinan rute meskipun perlu mengambil rute yang lain dan tidak memberikan rekomendasi rute terdekat untuk mencapai tempat tujuan. Sedangkan pada algoritma Dijkstra terlihat lebih konsisten dengan memilah rute yang jaraknya lebih dekat. Konsistensi ini yang menjadikan algoritma Dijkstra lebih banyak digunakan dalam penentuan jarak terpendek ataupun jarak terbaik jika dilihat berdasarkan jarak tempuh.

Hasil yang dicapai pada penelitian ini berupa aplikasi pencarian rute terbaik untuk mencapai tujuan lokasi parkir yang dituju dengan menerapkan algoritma Dijkstra. Dalam penerapan aplikasi android, menampilkan rute terbaik untuk mencapai sebuah tempat parkir di gambarkan dengan garis rute (*polyline*) yang berwarna lebih terang (merah) di banding dengan alternatif rute lainnya (terlihat dalam Gbr.12).

V. KESIMPULAN DAN SARAN

Setelah melakukan perancangan pada Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada Mobile E-Parking Berbasis Sistem Informasi Geografis, maka dapat disimpulkan bahwa peneliti telah mampu membuat sistem yang dirancang untuk menyediakan informasi parkir berbasis Sistem Informasi Geografis pada platform android, dan menyediakan navigasi rute terbaik dengan menerapkan Algoritma Dijkstra pada fitur Google Maps.

Adapun kemungkinan pengembangan selanjutnya mengenai fitur yang lebih kompleks lagi saat melakukan pencarian rute ke tempat parkir tujuan yakni: informasi jarak setiap kemungkinan rute yang akan dilalui dan keadaan lalu lintas yang dapat di akses menggunakan fitur Google Maps.

Saran yang ingin disampaikan untuk pengembangan perancangan pada penelitian ini adalah perlu pengembangan dalam menangani kelemahan algoritma Dijkstra yang hanya berfokus pada menghitung jarak antara titik awal sampai titik tujuan tanpa memperhatikan faktor lain seperti kemacetan yang terjadi pada jalan, waktu tempuh dan faktor lain yang menyebabkan pengguna parkir akan terlambat sampai ke tempat tujuan. Dalam pengembangan sistem diperlukan sistem reservasi untuk melakukan booking jarak jauh tempat

parker. Dalam penerapannya diupayakan adanya keuntungan dari sistem reservasi sehingga pengguna parkir akan mengutamakan memesan terlebih dahulu menggunakan aplikasi dibanding memesan secara langsung dimana pengguna parkir tidak mengetahui apakah tempat parkir tujuan sudah penuh atau tidak.

UCAPAN TERIMA KASIH

Terimakasih kepada rekan-rekan di Politeknik Pos Indonesia khususnya program studi D4 Teknik Informatika yang dalam penelitian banyak memberikan bantuan, bimbingan dan dorongan sehingga penelitian ini dapat terlaksana.

DAFTAR PUSTAKA

- [1] S. Greco, C. Molinaro, and C. Pulice, "Efficient maintenance of shortest distances in dynamic graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 474–487, 2018.
- [2] Y. Wang, X. Li, and R. Ruiz, "An exact algorithm for the shortest path problem with position-based learning effects," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 11, pp. 3037–3049, 2017.
- [3] S. Bhattacharjee, P. Mitra, and S. K. Ghosh, "Spatial interpolation to predict missing attributes in gis using semantic kriging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4771–4780, 2014.
- [4] S. Ye, D. Zhu, X. Yao, N. Zhang, S. Fang, and L. Li, "Development of a highly flexible mobile gis-based system for collecting arable land quality data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 11, pp. 4432–4441, 2014.
- [5] S. Zhu and D. Levinson, "Do people use the shortest path? an empirical test of wardrop's first principle," *PloS one*, vol. 10, no. 8, p. e0134322, 2015.
- [6] S. Trilles, O. Belmonte, L. D'iaz, and J. Huerta, "Mobile access to sensor networks by using gis standards and restful services," *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4143–4153, 2014.
- [7] K. J. Kearfott, Z. D. Whetstone, and K. M. Rafique Mir, "Use of a geographic information system (gis) for targeting radon screening programs in south dakota," *Journal of radiation research*, vol. 57, no. 1, pp. 84–90, 2015.
- [8] C. Zinoune, P. Bonnifait, and J. Ibañez-Guzmán, "Sequential fdia for autonomous integrity monitoring of navigation maps on board vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 143–155, 2016.
- [9] A. B. Marlintha, W. Java, B. Irawan, W. Java, R. Latuconsina, and W. Java, "Design and implementation of smart village mapping geographic information system based web in the cinunuk village," pp. 66–71, 2017.
- [10] O. Lumasuge, "Implementation Analytic Network Process Method and Geographic Information System To Determine The Freshwater Fish Farming Location," pp. 107–112, 2017.
- [11] S. Adi and J. Suhartono, "Smart Village Geographic Information System (GIS) Development In Indonesia And Its Analogous Approaches," no. November, pp. 65–70, 2017.
- [12] Z. Cao, H. Guo, J. Zhang, D. Niyato, and U. Fastenrath, "Finding the shortest path in stochastic vehicle routing: A cardinality minimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1688–1702, 2016.
- [13] S. Ma, K. Feng, J. Li, H. Wang, G. Cong, and J. Huai, "Proxies for shortest path and distance queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1835–1850, 2016.
- [14] Y. Zhang, Y. Su, J. Yang, J. Ponce, and H. Kong, "When dijkstra meets vanishing point: a stereo vision approach for road detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–12, 2018.
- [15] J.-d. Zhang, Y.-j. Feng, F.-f. Shi, G. Wang, B. Ma, R.-s. Li, and X.-y. Jia, "Vehicle routing in urban areas based on the oil consumption weight-dijkstra algorithm," *IET Intelligent Transport Systems*, vol. 10, no. 7, pp. 495–502, 2016.
- [16] A. Sedeno-Noda and A. Raith, "A dijkstra-like method computing all extreme supported non-dominated solutions of the biobjective shortest path problem," *Computers & Operations Research*, vol. 57, pp. 83–94, 2015.
- [17] O. G. Swathika and S. Hemamalini, "Prims-aided dijkstra algorithm for adaptive protection in microgrids," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 4, no. 4, pp. 1279–1286, 2016.